Task 2

1. write a command to create a file named demo.txt inside Movies directory.

**touch Movies/demo.txt**

This command creates an empty file named "demo.txt" within the "Movies" directory. If the "Movies" directory doesn't exist, touch will not create it for you.

1. copy hello.txt files from Desktop to Downloads directory.

**cp ~/Desktop/hello.txt\* ~/Downloads/**

cp: The command to copy files.

~/Desktop/hello.txt\*: The source path.

* ~: Represents your home directory (where Desktop is typically located).
* /Desktop/: The Desktop directory path.
* hello.txt\*: The wildcard \* matches any number of characters after "hello.txt", so it copies all files named "hello.txt" (or "hello.txt1", "hello.txt2", etc.).

~/Downloads/: The destination path, your Downloads directory.

1. Write a command to display all the files from Movies Directory

**ls Movies/**

This command breaks down as follows:

* ls: The ls command is used to list the contents of a directory.
* Movies/: This specifies the directory you want to list the contents of. The trailing slash (/) is important to indicate that it's a directory and not a file.

This command will display all files (including hidden files starting with ".") within the "Movies" directory.

**Options for ls:**

* ls -l Movies/: This displays a long listing format with detailed information about each file (permissions, owner, size, etc.).
* ls -a Movies/: This displays all files, including hidden ones

.

1. . display first 15 lines of demo.txt file

**head -n 15 Movies/demo.txt**

This command combines two commands:

* head: This command is used to display the first few lines of a file.
* -n 15: This option specifies the number of lines to display (15 in this case).
* Movies/demo.txt: This is the path to the file you want to display the first lines from.

**Important note:**

* Make sure the Movies directory exists and the demo.txt file is located within it. Otherwise, the command will return an error message.

E. using cat command, create a new file and write the data to the file.

**cat > new\_file.txt**

This command works by:

1. cat: Invokes the cat command.
2. >: Redirects the standard input of cat to a file.
3. new\_file.txt: Specifies the name of the file to be created (replace with your desired filename).

**Here's the catch:**

* After running this command, your terminal will enter an interactive mode.
* Type the data you want to write to the file.
* To save and exit, press Ctrl+D (on most systems).

**Example:**

Bash

cat > my\_data.txt

This is some data I want to write to the file.

I can add multiple lines here.

^D // Press Ctrl+D to save and exit

**Explanation:**

* The lines you type after cat > my\_data.txt will be written to the file.
* Pressing Ctrl+D indicates the end of input and saves the data to my\_data.txt.

**Alternative approach using echo:**

Bash

echo "This is some data for the file." > my\_data.txt

Use code with caution.

content\_copy

This approach uses echo to create a string and redirects it to the file using >. It's useful for writing a single line of data.

**Remember:**

* Both methods create the file if it doesn't exist.
* They overwrite the existing content if the file already exists.

F. read the file content of demo.txt

**1. Using cat:**

Bash

cat Movies/demo.txt

Use code with caution.

content\_copy

This is the simplest approach. The cat command displays the entire content of the file on the terminal.

**2. Using more or less (for long files):**

Bash

more Movies/demo.txt

Use code with caution.

content\_copy

Bash

less Movies/demo.txt

Use code with caution.

content\_copy

These commands are useful for displaying the contents of long files one screen at a time.

* more: Displays one page of content at a time. Press space for next page, b for previous page, and q to quit.
* less: Similar to more but allows scrolling up and down using the arrow keys.

**3. Using redirection with a shell script (optional):**

Create a script named read\_file.sh (or any name you prefer) with the following content:

Bash

#!/bin/bash

content=$(cat Movies/demo.txt)

echo "File content:"

echo "$content"

Use code with caution.

content\_copy

* Make the script executable:

Bash

chmod +x read\_file.sh

Use code with caution.

content\_copy

* Run the script:

Bash

./read\_file.sh

Use code with caution.

content\_copy

This script:

1. Defines a variable content to store the file content using cat.
2. Prints a message "File content:".
3. Prints the variable content which holds the file data.

**Important note:**

* Replace Movies/demo.txt with the actual path to your file if it's located in a different directory.
* Ensure you have read permissions for the demo.txt file.